CD-MAKE 2017

The more the merrier - Federated learning from local sphere recommendations

Bernd Malle, Nicola Giuliani, Peter Kieseberg and Andreas Holzinger

b.malle@hci-kdd.org

Introduction and Motivation

- Thinking about Machine learning from a *European* startup perspective

- Which brings a few particular challenges with it:
  - Usually less startup capital than U.S. competitors
    => therefore less money for computing power
  - much fewer possible customers (initially) than Asian competitors
    => less initial data
  - GDPR is a major impediment
    => expressely prohibits use of personal data...

- Maybe we can circumnavigate all those hurdles via
  Client-side Machine Learning

- How far away are relevant decision points within a social networks usually?
- Leskovec, even back in 2006 observed recommender cascades within an online shopping system
  - for 3 out of 4 products: maximum size of cascade was < 10

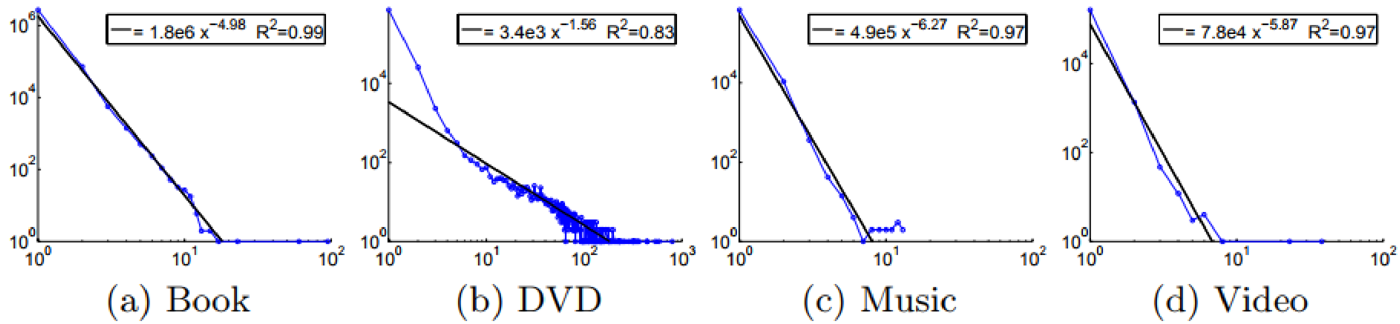


Fig. 1. Size distribution of the cascades for the four product types (log size of cascade vs. log count). Superimposed line presents a power-fit. R^2 is the coefficient of determination.

Leskovec, Jure, Ajit Singh, and Jon Kleinberg. "Patterns of influence in a recommendation network." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2006.

Recommendation Cascades

- most were not chains, but one node influencing many others (splits) or several recommendations directed at one node (merges)
- single recommendations made up the majority of 'cascades'
- overall, the average ego network from which relevant recommendations originated was little more than 1(!)

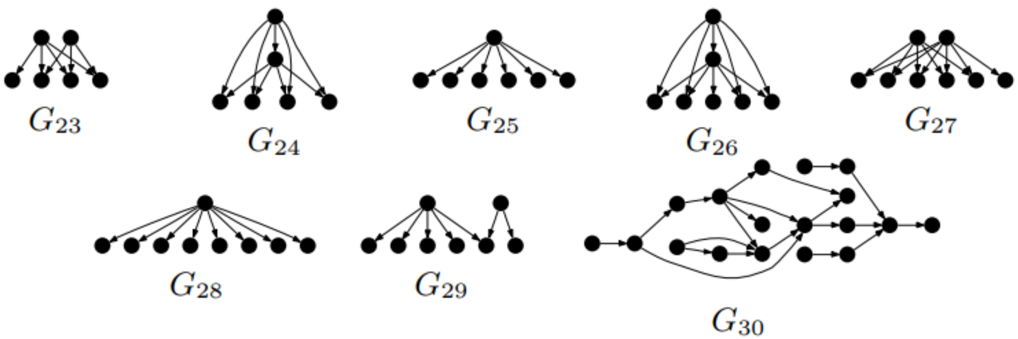


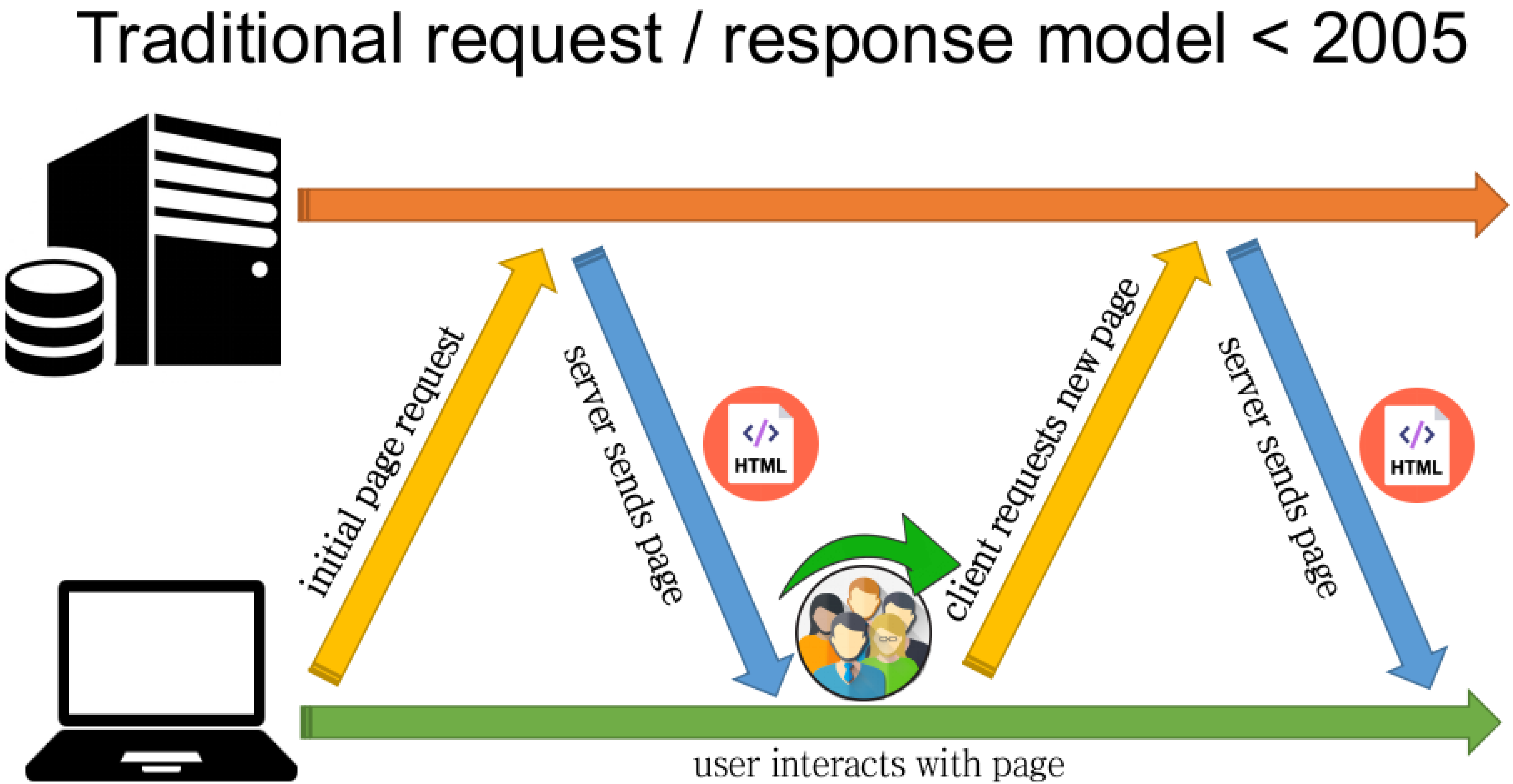
Fig. 2. Typical classes of cascades. G_{23}, G_{27}: nodes recommending to the same set of people, but not each other. G_{24}, G_{26}: one node recommends to another, and both recommend to the same community. G_{25}, G_{28}, G_{29}: a flat cascade. G_{30} is an example of a large cascade.

Leskovec, Jure, Ajit Singh, and Jon Kleinberg. "Patterns of influence in a recommendation network." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2006.

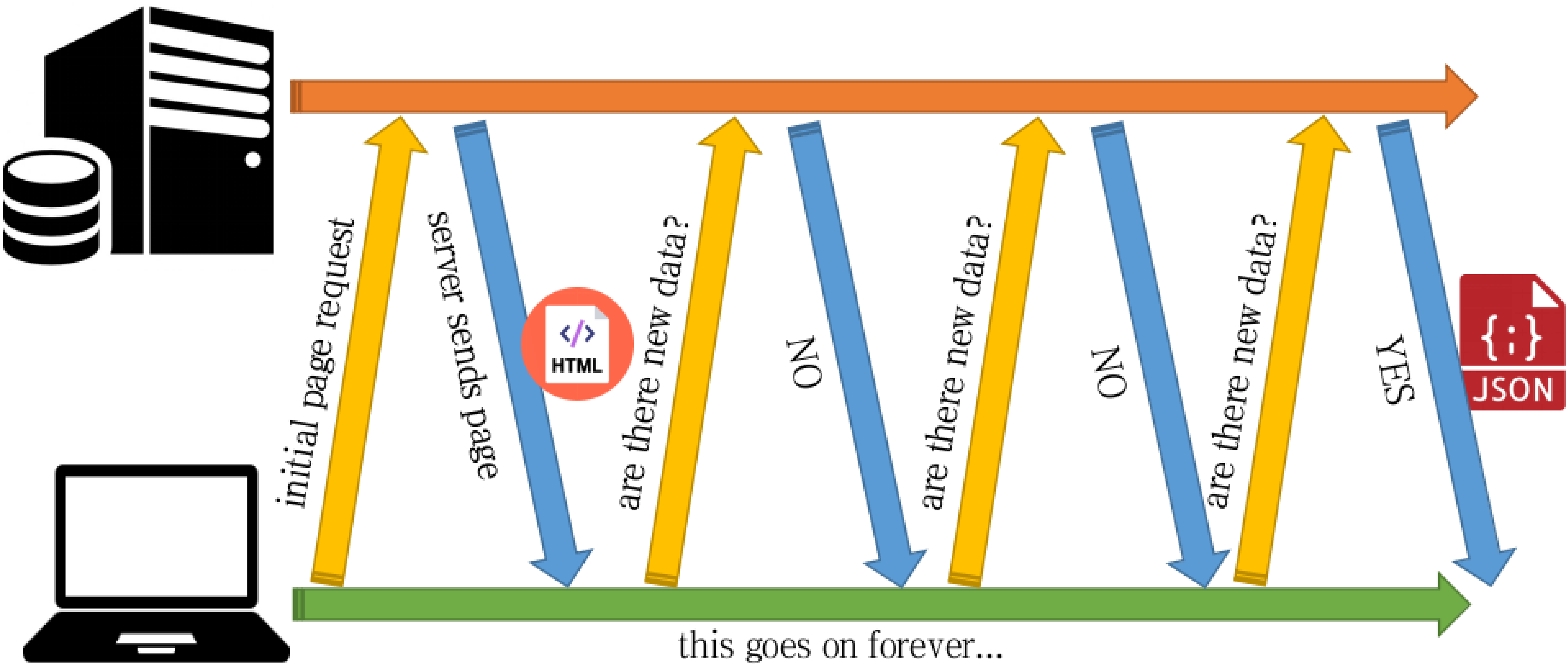
Introduction and Motivation

- the FB graph has been estimated to have a diameter of as low as 4
- the diameter is shrinking with new connections (despite new users)
- although the graph is globally very sparse, individual node neighborhoods contain surprisingly dense structure
- Conclusion: We dont need the whole graph to calculate good recommendations – in most cases we only need to take a node's immediate vicinity into account!

Traditional request / response cycle

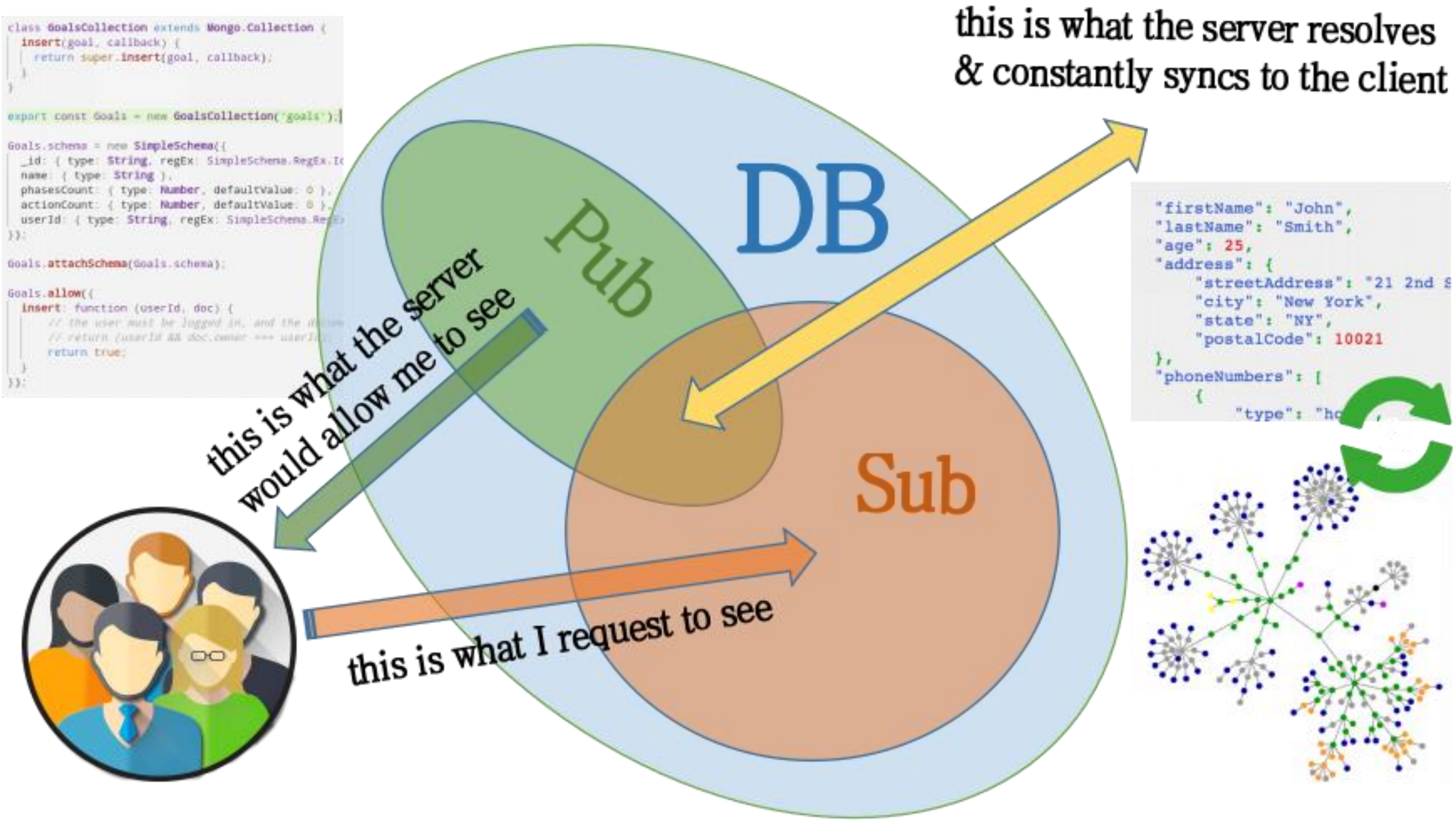


Longpolling via Ajax 2005 - 2012 (& still in use)



Publish / Subscribe with automatic updates

Modern Pub/Sub with constant synchronization



Consequences of Pub/Sub

- This means in effect, that all information within the neighborhood of a node (if you see it from a network perspective) is constantly available within the browser / mobile device
  - my direct friends on a social network
  - the information / resources within my project group
- Combining the two principles of Pub/Sub and recommender cascades, we see that a majority of relevant recommend-dations could be computed directly on the client

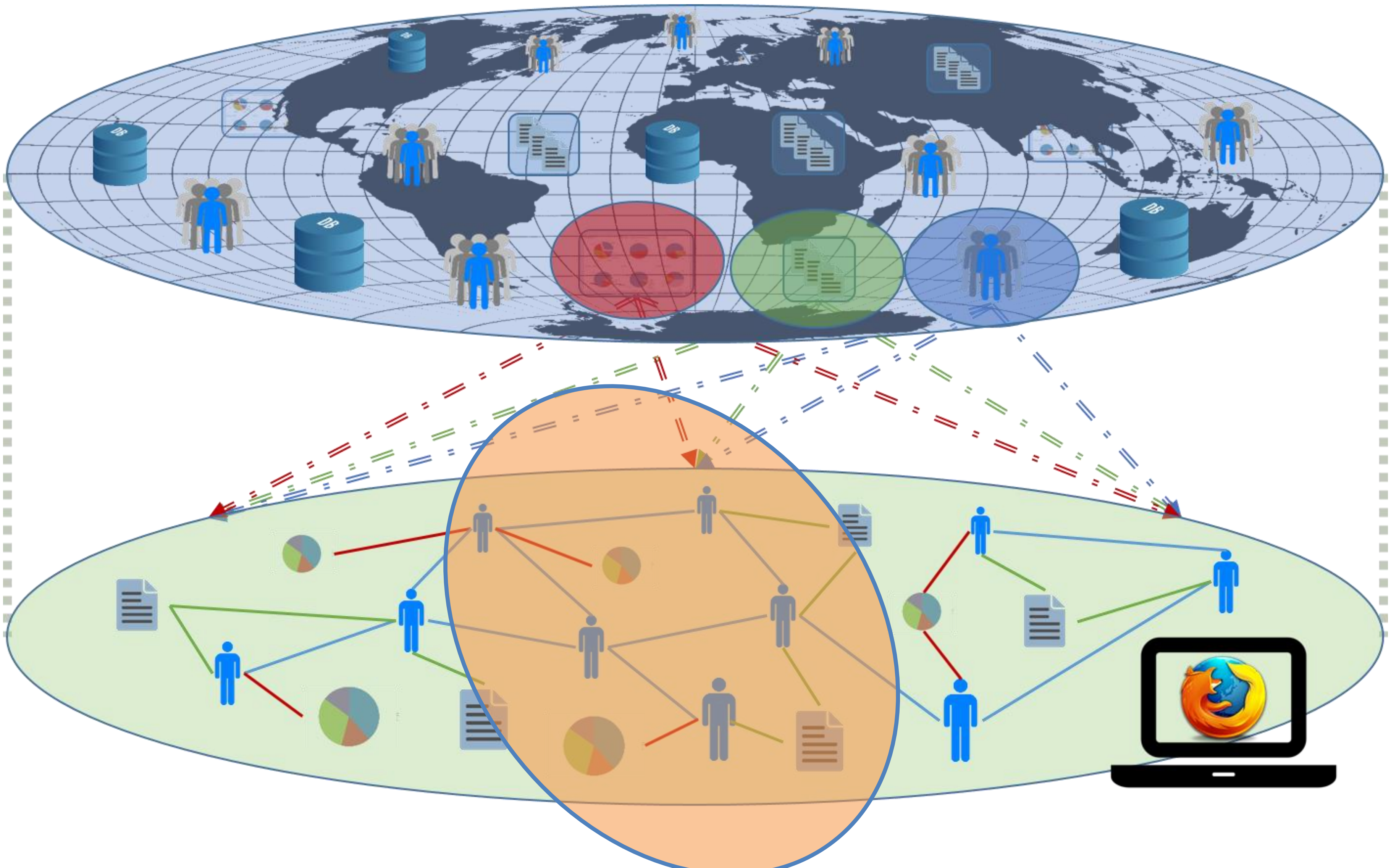
Global sphere / Local sphere

- Many globally distributed databases *dont have to / cannot be* implemented as a graph (facebook also does not store one central graph, but rather superimposes relational DBs with a logical graph layer - GraphQL)
  - would need too many globally propagating updates – "the consequences of a tiny node update could affect the farthest reaches of the global graph"

- The local sphere can get it's information from many pub/sub mechanism targeting different endpoints in the global sphere

- The local sphere is a superset of the actual user's data enriched with data from it's relevant vicinity, plus possibly sensitive information only available on the client itself.

Global Sphere / Local Sphere / User data



GraphQL / Websockets / etc.

Advantages

- GDPR says processing of personal data is expressely prohibited
  - but they are talking about data you **collected**
  - what if you haven't collected it because it never left the users' device?

- You can potentially use a wealth of information available on the client device you could never access server-side
  - address book
  - calendar
  - GPS
  - emails
  - messaging services
  - .....

Advantages (coming back to the startup idea)

- BYOPP - "Bring your own processing power"
  - World's fastest supercomputer can do 93 Petaflops
  - A Geforce 1080 runs at about 12 Teraflops (single prec.)
  - You need ~7,750 customers with such cards to reach the TaihuLight
  - iPhone 7: GPU operates at 729.6 GFLOPS
  - ~130k iphones stack up to the TaihuLight

- of course apps can never access the full potential of a client device (think of battery alone…), but those numbers give a good feeling about the magnitude we're talking about

- a few hundred thousand users is not much for a successful startup today => SCALABILITY !!!

Proposed mechanism

1. Pub/Sub keeps the local sphere in sync with global data

2. Local algorithms compute recommendations
   - these can also come from a client-side crawler (like the FB crawler which scans URLs you paste into a comment field and extracts images from a website)

3. Upon user acceptance, a new node is introduced into the local sphere + updated to the global sphere

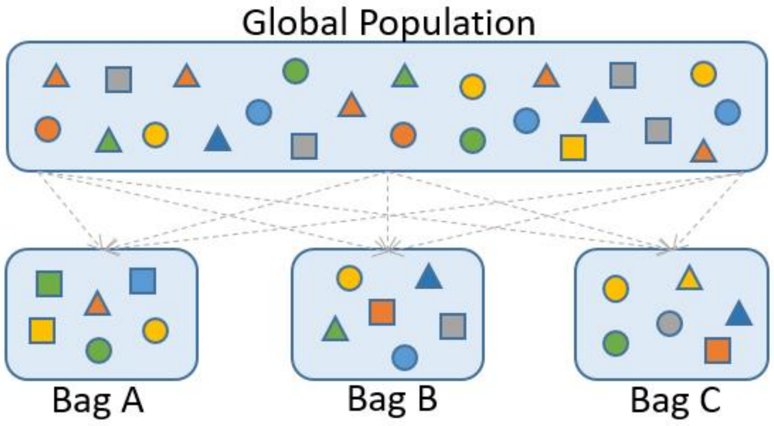
4. Client devices with overlapping local spheres now receive that node in the background

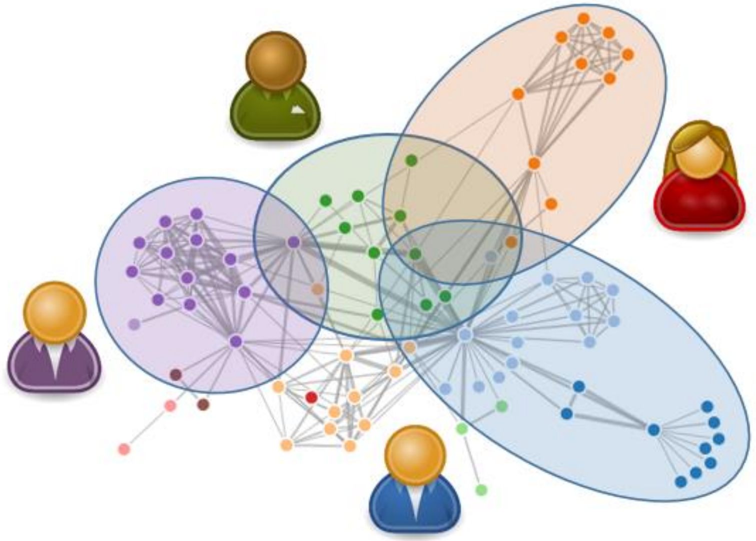
5. Their recommenders respond… => 1…

Machine Learning / Conclusion

- Maybe it's even possible to implement Machine Learning paradigms on such a distributed platform (see bagging below)

- Eventually, we might not even have to curate a global graph anymore - it could be a virtual, implicit, query-able instantiation of the sum of all local spheres at any given point in time!





Thank you!

Bernd Malle <b.malle@hci-kdd.org>

Reggio Calabria, 2017-09-01